

Table of Contents

CGI and Command Line Scripts.....	1
CGI Scripts.....	1
General Information.....	1
CGI environment.....	1
Command-line.....	1
Common parameters.....	1
attach.....	2
changes.....	2
configure.....	2
edit.....	2
login.....	3
logon.....	3
manage.....	3
action=create.....	3
action=createweb.....	4
action=editSettings.....	4
action=saveSettings.....	4
action=bulkRegister.....	4
action=changePassword.....	4
action=resetPassword.....	5
action=deleteUserAccount.....	5
action=restoreRevision.....	5
oops.....	5
preview.....	6
rdiff.....	6
register.....	6
rename.....	6
action="renameweb".....	6
action=anything else.....	7
rest.....	7
Invocation Examples.....	7
Retrieving passed values.....	8
save.....	8
search.....	9
statistics.....	11
upload.....	11
view.....	12
viewfile.....	12
Command Line Scripts.....	13
geturl.pl.....	13
rewriteshebang.pl.....	13
tick_foswiki.pl.....	13

CGI and Command Line Scripts

Programs on the server performing actions such as rendering, saving and renaming topics.

These scripts are located in the `bin` and `tools` directories. This topic describes the interfaces to some of those scripts. All scripts in the `bin` directory can be called from the CGI (Common Gateway Interface) environment or from the command line. The scripts in the `tools` directory can only be called from the command line.

CGI Scripts

Details on CGI scripts located in the `bin` directory.

General Information

CGI environment

In the CGI environment parameters are passed to the scripts via the URL and URL parameters. Environment variables are also used to determine the user performing the action. If the environment is not set up, the default user is used (usually `guest`).

Command-line

You **must** have the `bin` directory on the perl path to run the scripts from the command line. To avoid issues with file permissions, run the scripts as the web server user such as `nobody` or `www`.

Parameters are passed on the command line using '-name' - for example,

```
$ cd /usr/local/foswiki/bin
$ save -topic MyWeb.MyTopic -user admin -action save -text "New text of the topic"
```

All parameters require a value, even if that is the empty string.

Common parameters

All the scripts accept a number of common parameters. The first two components of the URL after the script name are taken as the web and the topic, respectively. Standard URL parameters are:

Parameter	Description	Default
topic	If this is set to a URL, Wiki will immediately redirect to that URL. Otherwise it overrides the URL and is taken as the topic name (you can pass <code>Web.TopicName</code>)	
user	Command-line only; set the name of the user performing the action. Note: this usage is inherently insecure, as it bypasses webserver login constraints. For this reason only authorised users should be allowed to execute scripts from the command line.	
skin	Overrides the default skin path (see Skins)	
cover	Specifies temporary skin path to prepend to the skin path for this script only (see Skins)	
t	While the <code>t</code> parameter is not actively used by any scripts, it is used throughout the links like <code>edit</code> to ensure that the browser doesn't cache the reply.	generally set to current time
logout	requests the <code>LoginManager?</code> to log the current user out. (happens at the begining of the request so will terminate any other operation requested)	

attach

Despite the name, this script doesn't actually attach a file to a topic - for that, use `upload`. This script is part of the transactions sequence executed when a file is uploaded from the browser. it just generates the "new attachment" page for a topic.

Parameter	Description	Default
<code>filename</code>	Name of existing attachment (if provided, this is a "manage attachment" action)	none (in which case this is a "new attachment" action)

changes

Shows all the changes in the given web.

The `changes` script can receive one parameter:

Parameter	Description	Default
<code>minor</code>	If 0, show only major changes. If 1, show all the changes (both minor and major)	0

The main difference between invoking this script and using `WebChanges` is that `WebChanges` is based on a `%SEARCH%`, while this script reads the `changes` file in each web, making it much faster.

NOTE: The result from `changes` script and the topic `WebChanges` can be different, if the `changes` file is deleted from a web. In particular, in new installations the `changes` script will return no results while the `WebChanges` topic will.

configure

`configure` is the browser script used for inspection of, and changes to, the site configuration. None of the parameters to this script are useable for any purpose except `configure`. See `configure`.

edit

The `edit` script understands the following parameters, typically supplied by HTML input fields:

Parameter	Description	Default
<code>action</code>	Optional. Use the editaction template instead of the standard edit. If <code>action=text</code> , then hide the form. If <code>action=form</code> hide the normal text area and only edit the form. You can change the Edit/Edit Raw buttons to always append the action parameter in skins like Pattern and Classic by setting the topic or preference setting <code>EDITACTION</code> to the value <code>text</code> or <code>form</code> . To edit the topic once the <code>EDITACTION</code> is defined as <code>form</code> simply remove the <code>action=form</code> from the browser URL of the edit script and reload the edit window	
<code>onlynewtopic</code>	If set, error if topic already exists	
<code>onlywikiname</code>	If set, error if topic name is not a WikiWord	
<code>templatetopic</code>	The name of the template topic, copied to get the initial content (new topic <i>only</i>)	
<code>text</code>	Initial text for the topic	
<code>topicparent</code>	The parent topic	
<code>formtemplate</code>	Name of the form to instantiate in the topic. Overrides the form set in the <code>templatetopic</code> if defined. (will remove the form is set to 'none')	
<code>contenttype</code>	Optional parameter that defines the application type to write into the CGI header. Defaults to <code>text/html</code> . May be used to invoke alternative client	

Parameter	Description	Default
	applications	
anyname	Any parameter can passed to the new topic; if the template topic contains %URLPARAM{ "anyname" }%, it will be replaced by its value	
breaklock	If set, any lease conflicts will be ignored, and the edit will proceed even if someone is already editing the topic.	
redirectto	If the user continues from edit to save, and if the save (or cancels the edit) process is successful, save will redirect to this topic or URL. The parameter value can be a TopicName, a Web.TopicName, or a URL. Note: Redirect to a URL only works if it is enabled in configure (Miscellaneous {AllowRedirectUrl}).	

Form field values are passed in parameters named 'field' - for example, if I have a field `Status` the parameter name is `Status`.

1. The first sequence of ten or more X characters in the topic name will be converted on save to a number such that the resulting topic name is unique in the target web.

NOTE: most skins support the definition of `EDIT_SKIN`, which is used as the value of the `cover` parameter in edit URLs. This allows you to override the default edit skin on a web, topic or user basis.

login

Used for logging in with `TemplateLoginManager`.

Parameter	Description	Default
origurl	URL that was being accessed when an access violation occurred. the login process will redirect to this URL if it is successful	none
username	username of user logging in	none
password	password of user logging in	none

logon

Used for logging in when Web Server authentication is being used (e.g. `ApacheLoginManager`). The script does nothing; it is purely a placeholder for triggering the login process. The webserver will be set up to require a valid user to access this script, thus triggering the webserver login process.

manage

Performs a range of management functions.

 **Note:** The `manage` script can only be called via HTTP POST method. Make sure you specify `method="post"` if you call the `manage` script via a form action. It is not possible to call `manage` from an `<A href` link.

Parameter	Description	Default
action	One of <code>create</code> , <code>createweb</code> , <code>changePassowrd</code> , <code>resetPassword</code> , <code>bulkRegister</code> , <code>deleteUserAccount</code> , <code>editSettings</code> , <code>saveSettings</code> , <code>restoreRevision</code>	none

action=create

Alternative entry point for creation, via `edit`, of a new topic, used by screens that support several actions using `manage`.

edit

Parameter	Description	Default
topic	Name of topic to create (can be web.topic)	none

Other parameters are the same as for edit.

action=createweb

Create a new web

Parameter	Description	Default
newweb	Name of the new web	"
baseweb	Name of the web to copy to create the new web	"
webbgcolor	value for WEBBGCOLOR	"
sitemapwhat	Value for SITEMAPWHAT	"
sitemapusetto	Value for SITEMAPUSETO	"
nosearchall	Value for NOSEARCHALL	"
newtopic	Value of %TOPIC% within the web creation message. Optionally used in some skins to signify a non-default home topic.	

action=editSettings

No parameters

action=saveSettings

Parameter	Description	Default
text	Text of the topic	"
originalrev	Revision that the edit started on	Most recent revision
redirectto	If the savesettings process is successful, save will redirect to this topic or URL. The parameter value can be a TopicName, a Web.TopicName, or a URL. Note: Redirect to a URL only works if it is enabled in configure (Miscellaneous {AllowRedirectUrl}).	

All other parameters may be interpreted as form fields, depending on the current form definition in the topic.

action=bulkRegister

See BulkRegistration.

Parameter	Description	Default
OverwriteHomeTopics	Whether to overwrite existing home topics or not	false
EmailUsersWithDetails	Whether to mail registered users or not	false
LogTopic	Topic to save the log in	Same as topic name, with 'Result' appended.

action=changePassword

Change password, email address, or both, of a user.

Parameter	Description	Default
username	god alone knows	none
oldpassword	current password	none

action=create

Parameter	Description	Default
password	new password	<i>none</i>
passwordA	new password confirmation	<i>none</i>
email	new email address	<i>none</i>

password, =passwordA and email are optional. If neither or password and passwordA is set, then the user password is left unchanged. If email is unset, their email is left unchanged.

action=resetPassword

Reset the password for a single or multiple users

Parameter	Description	Default
LoginName	list of usernames to reset	none - error if not set
Introduction	message to be sent alongside the reset, most often used to announce to the user that they have been given an account.	"

This is used by BulkResetPassword and ResetPassword. Only administrators can provide a list of LoginNames, non-admins can only provide a single LoginName.

BulkRegistration provides the means to create multiple accounts but it does not announce those accounts to the users who own them. BulkResetPassword is used to assign the passwords, the Introduction is used to explain why they are receiving the mail.

action=deleteUserAccount

Unregisters (removes) the currently logged-in user.

Parameter	Description	Default
password	Users' password	none

action=restoreRevision

Alternative entry point for edit, used by screens that support several actions using manage . Parameters are as for =edit.

oops

This script is mainly used for rendering pages containing error messages, though it is also used for some functional actions such as manage pages (move topic etc).

oops templates are used with the oops script to generate system messages. This is done to make internationalisation or other local customisations simple.

The oops script supports the following parameters:

Parameter	Description	Default
template	Name of the template file to display	
def	Optional, can be set to the name of a single definition within template. This definition will be instantiated in the template wherever %INstantiate% is seen. This lets you use a single template file for many messages. For an example, see oopsmanagebad.tmpl.	
paramN	Where N is an integer from 1 upwards. These values will be substituted into template for %PARAM1% etc.	

action=changePassword

preview

This script is *deprecated*. Its functions are covered by the `save` script.

rdiff

Renders the differences between version of a topic

Parameter	Description	Default
rev1	the higher revision	
rev2	the lower revision	
render	the rendering style {sequential, sidebyside, raw, debug}	DIFFRENDERSTYLE, sequential
type	{history, diff, last} history diff, version to version, last version to previous	diff
context	number of lines of context	

TODO:

- add a {word} render style

register

Parameter	Description	Default
action	register or verify or resetPassword or approve	

⚠ Note: The `register` script can only be called via the HTTP POST method except when the action is `verify`. Make sure you specify `method="post"` if you call the `register` script via a form action. It is not possible to call `register` from an `<A href` link. The `verify` action is an exception as it is used to verify registration by clicking a href link from an email.

rename

Used for renaming webs, topics and attachments.

Parameter	Description	Default
action	renameweb or anything else	
skin	skin(s) to use	
confirm	if defined, requires a second level of confirmation	
referring_topics	(internal use only) list of topics that refer to the web or topic being renamed	
redirectto	If the rename process is successful, rename will redirect to this topic or URL. The parameter value can be a <code>TopicName</code> , a <code>Web.TopicName</code> , or a URL. Note: Redirect to a URL only works if it is enabled in <code>configure</code> (<code>Miscellaneous {AllowRedirectUrl}</code>).	

action="renameweb"

Rename a web.

Parameter	Description	Default
newparentweb	new parent web name	existing parent
newsubweb	new web name	

action=anything else

Rename a topic or an attachment.

Parameter	Description	Default
newweb	new web name	
newtopic	new topic name	
attachment	Attachment to move	none
template	template for error when an attachment doesn't exist, deleteattachment for when deleting an attachment	
currentwebonly	if defined, searches current web only for links to this topic	
nonwikiword	if defined, a non-wikiword is acceptable for the new topic name	

 **Note:** The rename script can only be called via the HTTP POST method. Make sure you specify `method="post"` if you call the rename script via a form action. It is not possible to call rename from an `<A href` link.

rest

This REST (Representational State Transfer) script can be invoked via http in the same way as the other scripts (see **Invocation Examples**, below) to execute a function that is associated to a "subject" and a "verb" (see below). These functions are usually registered by plugins using the `Foswiki::Func::registerRESTHandler` method. The rest script will print the result directly to the browser unless the `endPoint` parameter is specified, in which case it will output a redirect to the given topic.

The rest script supports the following parameters:

username	If TemplateLogin, or a similar login manager not embedded in the web server, is used, then you need to pass a username and password to the server. The username and password parameters are used for this purpose.
password	See username
topic	If defined as the full name (including web) of a topic, then when the script starts up plugins will be passed this as the "current" topic. If not defined, then Main.WebHome will be passed to plugins.
endPoint	Where to redirect the response once the request is served, in the form "Web.Topic"

The function is free to use any other query parameters for its own purposes.

 The rest script should **always** require authentication in any site that has logins. Otherwise there is a risk of opening up major security holes. So make sure you add it to the list of authenticated scripts if you are using ApacheLogin.

Invocation Examples

The rest script assumes that it will be called with URL in the form:

`http://my.host/bin/rest/<subject>/<verb>`

where `<subject>` must be the WikiWord name of one of the installed Plugins, and the `<verb>` is the alias for the function registered using the `Foswiki::Func::registerRESTHandler` method. The `<subject>` and `<verb>` are then used to lookup and call the registered function.

`<subject>` and `<verb>` are checked for illegal characters exactly in the same way as the web and topic names.

As an example, the EmptyPlugin has registered a function to be used with the `rest` script under the subject **EmptyPlugin** and the verb **example**. Click below to see the `rest` script in action (run as guest).

Call the Plugin

Note that for Plugins to register REST handlers, they must be enabled in `configure`.

Retrieving passed values

Additional parameters can be recovered via the query object in the `$session`, for example with the url:

```
http://my.host/bin/rest/MyPlugin/update?web=foo
```

The url parameters can be processed using:

```
my $query = $session->{request};
my $web = $query->{param}->{web}[0];
```

save

The `save` script performs a range of save-related functions, as selected by the `action` parameter.

Parameter	Description	Default
<code>action_save=1</code>	default ; save, return to view, <code>dontnotify</code> is off	
<code>action_quietsave=1</code>	save, and return to view, <code>dontnotify</code> is on	
<code>action_checkpoint</code>	save and redirect to the edit script, <code>dontnotify</code> is on	
<code>action_cancel</code>	exit without save, return to view	
<code>action_preview</code>	preview edited text	
<code>action_addform</code>	Redirect to the "change form" page.	
<code>action_replaceform...</code>	Redirect to the "change form" page.	
<code>action_delRev</code>	Administrators only delete the most recent revision of the topic - all other parameters are ignored. You have to be an administrator to use this, and not all store implementations will support it.	
<code>action_repRev</code>	Administrators only replace the text of the most recent revision of the topic with the text in the <code>text</code> parameter. <code>text</code> must include embedded meta-data tags. All other parameters are ignored. You have to be an administrator to use this, and not all store implementations will support it.	
<code>onlynewtopic</code>	If set, error if topic already exists	
<code>onlywikiname</code>	If set, error if topic name is not a WikiWord	
<code>dontnotify</code>	if defined, suppress change notification	
<code>templatetopic</code>	Name of a topic to use as a template for the text and form (new topic <i>only</i>)	
<code>text</code>	New text of the topic	
<code>forcenewrevision</code>	if set, forces a revision even if Wiki thinks one isn't needed	
<code>topicparent</code>	If 'none' remove any current topic parent. If the name of a topic, set the topic parent to this.	
<code>formtemplate</code>	if defined, use the named template for the form (will remove the form is set to 'none')	
<code>editaction</code>	When action is <code>checkpoint</code> , add form or replace form. . . , this is used as the <code>action</code> parameter to the edit script that is redirected to after the save is complete.	

Parameter	Description	Default
originalrev	Revision on which the edit started.	
edit	The script to use to edit the topic when action is checkpoint	edit
editparams	The parameter string to use to edit the topic	
redirectto	The save process will redirect to this topic or URL if it is successful. (Typically this would be the URL that was being viewed when edit was invoked). The parameter value can be a TopicName, a Web.TopicName, or a URL. Note: Redirect to a URL only works if it is enabled in configure (Miscellaneous {AllowRedirectUrl}).	view topic being edited

Any errors will cause a redirect to an oops page.

The parameters are interpreted in according to the following rules.

1. The first sequence of ten or more X characters in the topic name will be converted to a number such that the resulting topic name is unique in the target web.
2. When the action is save, checkpoint, quietsave, or preview:
 1. The new text is taken from the text parameter, if it is defined,
 - ◇ otherwise it is taken from the templatetopic, if it is defined, (new topic *only*)
 - ◇ otherwise it is taken from the previous version of the topic, if any,
 2. The name of the new form is taken from the formtemplate, if defined
 - ◇ otherwise it is taken from the templatetopic, if defined, (new topic *only*)
 - ◇ otherwise it is taken from the previous version of the topic, if any,
 - ◇ otherwise no form is attached.
 3. The value for each field in the form is taken from the query, if it is defined
 - ◇ otherwise it is taken from the templatetopic, if defined, (new topic *only*)
 - ◇ otherwise it is taken from the previous version of the topic, if any,
 - ◇ otherwise it defaults to the empty string.

Merging is only enabled if the topic text comes from text and originalrev is > 0 and is not the same as the revision number of the most recent revision. If merging is enabled both the topic and the meta-data are merged.

Form field values are passed in parameters named 'field' - for example, if I have a field Status the parameter name is Status.

Note: The save script can only be called via HTTP POST method. Make sure you specify method="post" if you call the save script via a form action. Example:


```
<form name="new" action="%SCRIPTURLPATH{save}%/Sandbox/" method="post">
...
</form>
```

It is not possible to call save from an <A href link.

search

CGI gateway to the %SEARCH% functionality driven by the following CGI parameters:

Parameter:	Description:	Default:
search="text"	(Alternative to above)	N/A
bookview="on"	BookView search, e.g. show complete topic text. Very resource demanding. Use only with small result sets	Show entire topic content of found topics

Parameter:	Description: 	Default:
casesensitive="on"	Case sensitive search	Ignore case
web="Name" web="Main, Know" web="all"	Comma-separated list of webs to search. See Macros#VarSEARCH for more details.	Current web
header="..." format="..."	Custom format results: see FormattedSearch for usage & examples	Results in table
type="keyword" type="literal" type="regex"	Do a keyword search like <code>soap "web service" -shampoo</code> ; a literal search like <code>web service</code> ; or RegularExpression search like <code>soap;web service;!shampoo</code>	%SEARCHVAR-DEFAULTTYPE% preferences setting (literal)
nototal="on"	Do not show number of topics found	Show number
excludetopic="Web*" excludetopic="WebHome, WebChanges"	Exclude topics from search: A topic, a topic with asterisk wildcards, or a list of topics separated by comma.	None
expandvariables="on"	Expand embedded macros before applying a FormattedSearch on a search hit. Useful to show the expanded text, e.g. to show the result of a SpreadsheetPlugin <code>%CALC{ }%</code> instead of the formula	Raw text
nofinalnewline="on"	If on, the result of the macro does not end in a line by itself. Any text continuing immediately after the search macro on the same line will be rendered as part of the table generated by the search, if appropriate.	off
topic="WebPreferences" topic="*Bug"	Limit search to topics: A topic, a topic with asterisk wildcards, or a list of topics separated by comma.	All topics in a web
limit="all" limit="16"	Limit the number of results returned. This is done after sorting if <code>order</code> is specified	All results
date="..."	limits the results to those pages with latest edit time in the given time interval.	All results
separator=", "	Line separator between hits	Newline "\$n"
multiple="on"	Multiple hits per topic. Each hit can be formatted. The last token is used in case of a regular expression ";" and search	Only one hit per topic
reverse="on"	Reverse the direction of the search	Ascending search
"text"	Search term. Is a keyword search, literal search or regular expression search, depending on the <code>type</code> parameter. SearchHelp has more	required
scope="topic" scope="text" scope="all"	Search topic name (title); the text (body) of topic; or all (both)	"text"
nonoise="on"	Shorthand for <code>nosummary="on"</code> <code>nosearch="on"</code> <code>nototal="on"</code> <code>zeroresults="off"</code> <code>noheader="on"</code> <code>noempty="on"</code>	Off
nosummary="on"	Show topic title only	Show topic summary
order="topic" order="created" order="modified" order="editby" order=	Sort the results of search by the topic names, topic creation time, last modified time, last editor, or named field of DataForms. The sorting is done web by web; in case you want to sort across webs, create a formatted table and	Sort by topic name

Parameter:	Description:	Default:
"formfield(name) "	sort it with TablePlugin's initsort	
zeroreresults="off"	Suppress all output if there are no hits	zeroreresults="on", displays: "Number of topics: 0"
noempty="on"	Suppress results for webs that have no hits.	Show webs with no hits
noheader="on"	Suppress search header Topics: Changed: By:	Show search header
nosearch="on"	Suppress search string	Show search string

statistics

Refresh the WebStatistics topics in range of webs.

Parameter	Description	Default
webs	comma-separated list of webs to run stats on	all accessible webs
logdate	YYYYMM to generate statistics for	current month

for example:

1. from browser <https://wiki.iac.ethz.ch/bin/statistics> updates *all user webs*
2. from browser <https://wiki.iac.ethz.ch/bin/statistics?webs=Userweb,Sandbox> updates *Userweb,Sandbox*
3. from browser <https://wiki.iac.ethz.ch/bin/statistics/System> updates *System*
4. from command line `bin/statistics` updates *all user webs*
5. from command line `bin/statistics -webs=Userweb,Sandbox` updates *Userweb,Sandbox*
6. from command line `bin/statistics System.WebHome` updates *System*

see SiteTools#WebStatistics_site_statistics for updating statistics using cron.

upload

Uploads an attachment to a topic. The HTTP request is expected to be in `multipart/form-data` format.

Parameter	Description	Default
hidefile	if defined, will not show file in attachment table	
filepath	local (client) path name of the file being uploaded. This is used to look up the data for the file in the HTTP query.	
filename	deprecated, do not use	
filecomment	Comment to associate with file in attachment table	
createlink	if defined, will create a link to file at end of topic	
changeproperties	if defined, this is a property change operation only - no file will be uploaded.	null
redirectto	URL to redirect to after upload. {AllowRedirectUrl} must be enabled in configure. The parameter value can be a TopicName, a Web.TopicName, or a URL. Redirect to a URL only works if it is enabled in configure, and is ignored if noredirect is specified. Note: Redirect to a URL only works if it is enabled in configure (Miscellaneous {AllowRedirectUrl}).	
noredirect	Normally it will redirect to 'view' when the upload is complete, but also designed to be useable for REST-style calling using the 'noredirect' parameter. If this parameter is set it will return an appropriate HTTP status	

Parameter	Description	Default
	code and print a message to STDOUT, starting with 'OK' on success and 'ERROR' on failure.	

Tips

- You can use a tool like `curl` to upload files from the command line using this script.
- You can call upload easily from XMLHttpRequest in Javascript.

Note: The `upload` script can only be called via HTTP POST method. Make sure you specify `method="post"` if you call the `upload` script via a form action. It is not possible to call `upload` from an `<A href` link.

view

Used for viewing topics.

Parameter	Description	Default
<code>raw=on</code>	Shows the text of the topic in a scrollable textarea	
<code>raw=debug</code>	As <code>raw=on</code> , but also shows the metadata (forms etc) associated with the topic.	
<code>raw=text</code>	Shows only the source of the topic, as plain text (Content-type: text/plain). Only shows the body text, not the form or other meta-data.	
<code>raw=all</code>	Shows only the source of the topic, as plain text (Content-type: text/plain), with embedded meta-data . This may be useful if you want to extract the source of a topic to a local file on disc.	
<code>section</code>	Allows to view only a part of the topic delimited by a named section (see <code>VarSTARTSECTION</code>). If the given section is not present, no topic content is displayed.	
<code>contenttype</code>	Allows you to specify a different Content-Type : (e.g. <code>contenttype=text/plain</code>)	
<code>rev</code>	Revision to view (e.g. <code>rev=45</code>)	
<code>template</code>	Allows you to specify a different skin template, overriding the 'view' template the view script would normally use. The default template is <code>view</code> . For example, you could specify <code>/System/CommandAndCGIScripts?template=edit</code> . This is mainly useful when you have specialised templates for a Wiki Application.	
<code>topic</code>	redirects (at the beging of the cgi script running) to show the spcified Web.Topic, or, redirects to a URL, if allowed by <code>{AllowRedirectUrl}</code> and <code>{PermittedRedirectHostUrls}</code>	

Note: For historical reasons, the view script has a special interpretation of the `text` skin. This skin cannot be redefined.

viewfile

Used for viewing attachments. Normally, a site will publish the attachments (`pub`) directory using a URL. However if it contains sensitive information, you will want to protect attachments using `AccessControls`. In this case, you can use the `viewfile` script to give access to attachments while still checking access controls.

Parameter	Description	Default
<code>filename</code>	name of attachment	
<code>rev</code>	Revision to view	

Instead of using the `filename` parameter, you can append the attachment name to the end of the URL path (after the topic) e.g.

`https://wiki.iac.ethz.ch/bin/viewfile/Webname/TopicName/Attachment.gif`

Command Line Scripts

Details on command line scripts located in the `tools` directory.

`geturl.pl`

This is a very simple script to get the content of a web site. It is marked as *deprecated* and might be removed (or enhanced) in a future release. Its functions are covered by the standard `wget` and `curl` commands.

- Usage: `geturl.pl <host> <path> [<port> [<header>]]`
- Example: `geturl.pl some.domain /some/dir/file.html 80`
- Will get: `http://some.domain:80/some/dir/file.html`

`rewriteshebang.pl`

Simple script to rewrite the `#!/usr/bin/perl` shebang lines specific to your local Perl installation. It will rewrite the first line of all your cgi scripts so they use a different shebang line. Use it if your perl is in a non-standard location, or you want to use a different interpreter (such as 'speedy').

`tick_foswiki.pl`

This script executes a number of non-essential regular administration tasks that will help keep your site healthy and happy, such as removing expired sessions and lease files.

It is intended to be run as a cron job or a scheduled task once a week. Example crontab entry:

```
0 0 * * 0 cd /usr/local/foswiki/bin && perl ../tools/tick_foswiki.pl
```

Note: The script has to be run by a user who can write files created by the webserver user.

Related Topics: [AdminDocumentationCategory](#), [DeveloperDocumentationCategory](#)

[Edit](#) | [Attach](#) | [Print version](#) | [History: %REVISIONS%](#) | [Backlinks](#) | [Raw View](#) | [More topic actions](#)

Topic revision: [r1](#) - 12 Sep 2009 - 03:21:53 - [ProjectContributor](#)

- ☐ System

- Log In

- **Toolbox**

-  Users
-  Groups
-  Index
-  Search
-  Changes
-  Notifications
-  RSS Feed
-  Statistics
-  Preferences

- **User Reference**

- [BeginnersStartHere](#)
- [TextFormattingRules](#)
- [Macros](#)
- [FormattedSearch](#)

- QuerySearch
- DocumentGraphics
- SkinBrowser
- InstalledPlugins
- **Admin Maintenance**
 - Reference Manual
 - AdminToolsCategory
 - InterWikis
 - ManagingWebs
 - SiteTools
 - DefaultPreferences
 - WebPreferences
- **Categories**
 - Admin Documentation
 - Admin Tools
 - Developer Doc
 - User Documentation
 - User Tools
- **Webs**
 - ☐ Public
 - ☐ System

•
•



Copyright © by the contributing authors. All material on this site is the property of the contributing authors.

Ideas, requests, problems regarding Wiki? Send feedback